AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A system ~~to~~ that facilitates ~~facilitate~~ generalized comprehension in an imperative language, comprising:

a language component that receives a user input expression comprising comprehension notations, enabling ~~to enable~~ programming of comprehension notations in ~~an~~ the imperative language;

an interface component ~~to~~ that describes ~~describe~~ a meaning of the comprehension notations; and

a translation component ~~to~~ that analyzes the meaning of the comprehension notations and facilitates ~~facilitate~~ execution of the comprehension notations in accordance with the imperative language.

2. (Original) The system of claim 1, the language component includes a generalized comprehension that performs operations on a fixed or list comprehension.

3. (Original) The system of claim 1, the interface component defines one or more methods for the generalized comprehension.

4. (Original) The system of claim 1, the translation component includes at least one of just-in-time compilation techniques, interpretive techniques, and source code compilation techniques.

5. (Original) The system of claim 2, the language component enables users to define at least one of an implicit expression, an explicit expression, a mathematical expression, a

database expression, and a processing expression in accordance with the generalized comprehension.

6. (Currently amended) The system of claim 1, the comprehension notations comprise a mathematical parameterized monad ~~include at least one of the following syntax:~~

~~comprehension ::= type{ expression : qualifiers }~~

~~qualifiers ::= qualifier (, qualifier)*~~

~~qualifier ::= generator | filter | local-declaration~~

~~generator ::= type$_{opt}$ identifier in expression~~

~~filter ::= expression~~.

7. (Canceled) The system of claim 1, the interface component is associated with at least one of an IBuildable interface and an IBuilder interface.

8. (Currently amended) The system of claim [[7]] 1, wherein the interface component is associated with at least one of a final results function, an accumulation function for intermediate results, an early termination function, and a default value.

9. (Currently amended) The system of claim [[7]] 1, wherein the language, interface and translation components operate upon a generalized comprehension further comprising an assignment expression or a yield return statement.

10. (Canceled) The system of claim 7, further comprising an IEnumerable or IEnumerator component.

11. (Currently amended) The system of claim [[7]] 1, wherein the language, interface and translation components operate upon a generalized comprehension further comprising a relational database expression.

12. (Currently amended) The system of claim 11, wherein the relational database expression is employed for a query of a database.

13. (Original) A computer readable medium having computer readable instructions stored thereon for implementing the language component, the interface component, and the framework component of claim 1.

14. (Currently amended) A comprehension notation system, comprising:

means for defining a list comprehension set; ~~and~~

means for receiving a user input generalized comprehension expression, the generalized comprehension expression defined exterior to the list comprehension;

means for associating the list comprehension set with the generalized comprehension expression; and

means for providing an interface for the generalized comprehension expression.

15. (Original) The system of claim 14, further comprising means for compiling the list comprehension, the generalized comprehension expression, and the interface.

16. (Currently amended) A method for providing a programming environment for a user to input a generalized comprehension that is automatically translated into a language form in an imperative language, comprising:

defining a list comprehension expression; and

defining a generalized comprehension class as an exterior component to the list comprehension expression within an imperative language environment.

17. (Original) The method of claim 16, further comprising providing an interface class for the generalized comprehension class.

18. (Original) The method of claim 16, further comprising defining a results function for the interface class.

19. (Original) The method of claim 18, the results function returns a type that is at least one of similar and dissimilar to a type associated with the generalized comprehension class.

20. (Original) The method of claim 16, further comprising compiling the list comprehension expression and the generalized comprehension class to produce an executable format for the imperative language environment.

21. (Original) The method of claim 16, further comprising defining at least one relational database expression.

22. (Currently amended) A computer readable medium having a data structure stored ~~store~~ thereon, comprising:

a first data field that defines ~~to define~~ a static comprehension notation in the imperative language;

a second data field that defines ~~to define~~ a generalized comprehension notation received from a user input and comprising the static comprehension notation and a comprehension notation external to the static comprehension notation; and

a third data field that links ~~to link~~ the static comprehension notation with the generalized comprehension notation.

23. (Currently amended) The medium of claim 22, wherein the data structure further comprises ~~comprising~~ an interface field associated with the comprehension notation and at least one method associated with the interface field.

24. (Currently amended) The medium of claim 22, wherein the interface field is associated with at least one of a final results function, an accumulation function for intermediate results, an early termination function, and a default value.

25. (Currently amended) The medium of claim 22, wherein the generalized comprehension notation is associated with a user-defined expression.

26. (Canceled) The medium of claim 22, the generalized comprehension notation is associated with at least one of a `Sum`, `Average`, `Product`, `Forall`, `Exists`, `Choose`, Max, Seq, Multiset, Array, Add, and Query expression.

27. (Canceled) The medium of claim 26, further comprising a semantically equivalent expression.

28. (Original) The medium of claim 22, further comprising a comprehension type for a direct aggregation of collections.

29. (Original) The medium of claim 22, further comprising a field for an evaluation that is deduced from analyzing a portion of a collection.

30. (Original) The medium of claim 22, further comprising a field associated with at least one of a default value and an initialization value.

31. (Original) The medium of claim 22, further comprising a comprehension type that implicitly implements an interface.

32. (Original) The medium of claim 22, further comprising an interface pattern for defining aggregation functions on collections.

33. (New) The system of claim 6, wherein the parametrized monad comprehension notations comprises at least one selected from a group consisting of state, exceptions, parsers, non-determinism, and input-output expressions.

34. (New) The system of claim 6, wherein the parametrized monad comprehension comprises a Zermelo–Fraenkel set comprehension notation.